

Claims

1. Method for programming and/or executing programs for industrial automation systems, based on at least one computer unit with input aids, output aids and preferably at least one display device, with modules and functions respectively representing sub-tasks of an automation solution being modeled and/or created using the input aids and optionally the display device, with the modules and functions being structured and networked using the input aids and optionally the display device, so that they form at least one hierarchical tree as at least one machine-independent program, characterized in that the or each machine-independent program is loaded in the form of at least one hierarchical tree into the corresponding components of the automation system and that the corresponding components of the automation system execute the or each machine-independent program present in the form of at least one hierarchical tree.
2. Method according to claim 1, characterized in that the or each machine-independent program is executed on the corresponding components of the automation system with the aid of at least one object machine assigned to the same.
3. Method according to claim 1 or 2, characterized in that the or each machine-independent program, present in the form of at least one hierarchical object or operator tree in the corresponding components of the automation system, is processed interpretatively.
4. Method according to one or a plurality of claims 1 to 3, characterized in that the or each machine-independent program is present in the form of at least one object or operator tree

with a structure equivalent or similar to the representation of the program in the or each display device.

5. Method according to one or a plurality of claims 1 to 4,
5 characterized in that the or each machine-independent program is loaded into the corresponding components of the automation system using a machine-independent, symbolic representation of the or each hierarchical tree.

10 6. Method according to claim 5, characterized in that the machine-independent, symbolic representation of the or each hierarchical tree is in the form of a byte code or a markup language, in particular in the form of XML (Extended Markup Language).

15

7. Method according to one or a plurality of claims 1 to 6, characterized in that the or each object machine is configured as a real-time object machine with deterministic response and cycle times.

20

8. Method according to one or a plurality of claims 1 to 7, characterized in that the or each object machine provides operators, in particular mathematical and logical operators, and objects, in particular data objects and control objects,
25 from which the or each machine-independent program is formed in the form of the or each hierarchical tree.

9. Method according to claim 8, characterized in that the operators are instantiated during or after the loading of the
30 machine-independent program and the symbolic representation of the or each hierarchical tree is converted to physical addresses to generate a loadable program.

10. Method according to one or a plurality of claims 1 to 9, characterized in that the object machine is implemented as a function unit that is closed per se, which processes the or each hierarchical tree to the runtime.

5

11. Method according to one or a plurality of claims 1 to 9, characterized in that the object machine is implemented in a distributed manner as at least one object, with the or each hierarchical object or operator tree processing itself.

10

12. Method according to one or a plurality of claims 1 to 11, characterized in that the modules and functions, in particular objects, are assigned model information and/or meta-information using the input aids and optionally the display device.

15

13. Method according to one or a plurality of claims 1 to 12, characterized in that the objects of the machine-independent program present as a hierarchical object or operator tree are assigned a collection of infrastructure services or

20

infrastructure functions, which access the objects or the metadata assigned to the objects in a generic manner, so that an infrastructure service or infrastructure function can be used by all the objects and is applicable to all objects with metadata.

25

14. Device for programming and/or executing programs for industrial automation systems, based on at least one computer unit with input aids, output aids and preferably at least one display device, with means for modeling and/or creating modules and functions, which respectively represent the sub-tasks of an automation solution, and with means for structuring the modules and functions and for networking the same, to form at least one hierarchical tree as at least one machine-independent program,

30

characterized by means to load the or each machine-independent program in the form of at least one hierarchical tree into the corresponding components of the automation system, with the corresponding components of the automation system executing the
5 or each machine-independent program present in the form of at least one hierarchical tree.

15. Device according to claim 14, characterized by at least one object machine assigned to the corresponding components of
10 the automation system to execute the or each machine-independent program.

16. Device according to claim 14 or 15, characterized in that the or each machine-independent program is present in the form
15 of at least one object or operator tree with a structure equivalent or similar to the representation of the program in the or each display device.

17. Device according to claim 15 or 16, characterized in that
20 the or each object machine is configured as a real-time object machine with deterministic response and cycle times.

18. Device according to one or a plurality of claims 14 to 17, characterized in that the or each object machine provides
25 operators, in particular mathematical and logical operators, and objects, in particular data objects and control objects, from which the or each machine-independent program is formed in the form of the or each hierarchical tree.

30 19. Device according to one or a plurality of claims 14 to 18, characterized by means for assigning model information and/or meta-information to the modules and functions.

20. Device according to one or a plurality of claims 14 to 19, characterized in that the object machine is implemented as a function unit that is closed per se, which processes the or each hierarchical tree to the runtime.

5

21. Device according to one or a plurality of claims 14 to 19, characterized in that the object machine is implemented in a distributed manner as at least one object, with the or each hierarchical object or operator tree processing itself.

10

22. Device according to one or a plurality of claims 14 to 21, characterized in that the objects of the machine-independent program present as a hierarchical object or operator tree are assigned a collection of infrastructure services or

15

infrastructure functions, which access the objects via the containers assigned to the objects, so that an infrastructure service or infrastructure function can be used by all the objects.

20

23. Computer program implementing a method according to one or a plurality of claims 1 to 13 or a device according to one or a plurality of claims 14 to 22.

25

24. Data processing device, on which a computer program according to claim 23 is installed.